

## Method and system to handle inclusion of external files into a document processing language

**Patent number:** DE4313958

**Publication date:** 1993-11-04

**Inventor:** MOTOYAMA TETSURO (US)

**Applicant:** RICOH KK (JP)

**Classification:**

- international: **G06F17/22; G06F17/27; G06K15/00; G06F17/22; G06F17/27; G06K15/00; (IPC1-7): G06F15/20**

- european: **G06F17/22T; G06F17/22T2; G06F17/27; G06K15/00**

**Application number:** DE19934313958 19930428

**Priority number(s):** US19920876251 19920430

**Also published as:**



US5325484 (A1)



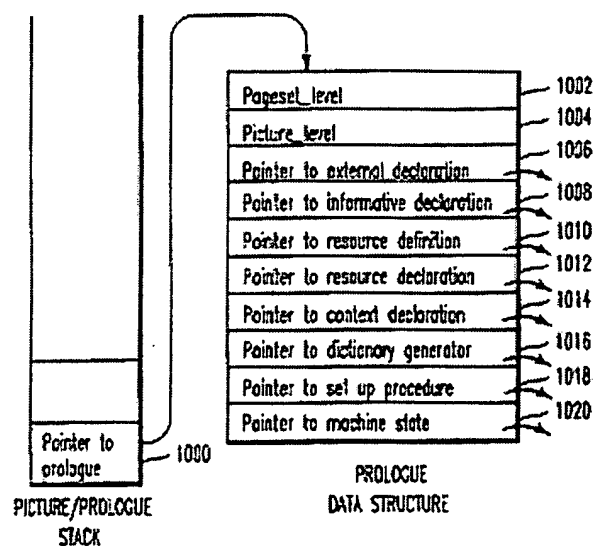
JP6083827 (A)

**Report a data error here**

Abstract not available for DE4313958

Abstract of corresponding document: **US5325484**

An apparatus for controlling printing or display of documents represented in a structured hierarchal page description language. Documents are provided as a document data stream which can include references to external references and external data declarations which can be incorporated into the document by the content processor. Documents are defined by a prologue section which may contain definitions and declaratory commands, with content portions containing specific tokens or command for specific images. The document data stream has a hierarchal structure. A means to provide external declarations is provided which will process external declarations as if the external declaration which can be stored on a separate device were part of the original document data structure. The storage of the references to the external declarations is accomplished in a tree linked stack structure with a last in first out arrangement which allows for a faster search order of external declarations. The tree linked stack structure further allows for the correct external declaration to be associated with a given reference thereto, even if different external declarations within the document data stream might have the same name. The context of the document data stream and the level in which a call is made determines the exact external declaration which is being referenced. This provides an incorporation by reference ability to a document data stream.



Data supplied from the esp@cenet database - Worldwide



①⑨ BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENTAMT

⑫ **Offenlegungsschrift**  
⑩ **DE 43 13 958 A 1**

⑤① Int. Cl.<sup>5</sup>:  
**G 06 F 15/20**

②① Aktenzeichen: P 43 13 958.2  
②② Anmeldetag: 28. 4. 93  
④③ Offenlegungstag: 4. 11. 93

DE 43 13 958 A 1

③⑩ Unionspriorität: ③② ③③ ③①  
30.04.92 US 876251

⑦① Anmelder:  
Ricoh Co., Ltd., Tokio/Tokyo, JP

⑦④ Vertreter:  
Schwabe, H., Dipl.-Ing.; Sandmair, K., Dipl.-Chem.  
Dr.jur. Dr.rer.nat.; Marx, L., Dipl.-Phys. Dr.rer.nat.,  
Pat.-Anwälte, 81677 München

⑦② Erfinder:  
Motoyama, Tetsuro, San Jose, Calif., US

Prüfungsantrag gem. § 44 PatG ist gestellt

⑤④ Einrichtung und Verfahren zum Steuern der Darstellung einer Vorlage

⑤⑦ Es ist eine Einrichtung zum Steuern des Druckens oder der Wiedergabe von Vorlagen geschaffen, welche in einer strukturierten hierarchischen Seitenbeschreibungssprache dargestellt sind. Die Vorlagen sind als ein Vorlagendatenstrom vorgesehen, welcher Verweise auf externe Referenzen und externe Datendeklarationen enthalten kann, welche in der Vorlage durch den Inhaltprozessor enthalten sein können. Vorlagen werden durch einen Prologabschnitt definiert, welcher Definitionen und erklärende Befehle enthalten kann, wobei Inhaltsteile spezifische Einheiten (Token) oder Befehle für spezifische Bilder enthalten. Der Vorlagendatenstrom hat eine hierarchische Struktur. Es ist eine Einrichtung vorgesehen, um externe Deklarationen zu schaffen, welche externe Deklarationen verarbeiten, wenn die externe Deklaration, welche als ein gesonderter Baustein gespeichert sein kann, Teil der ursprünglichen Vorlagendaten-Struktur war. Das Speichern der Verweise auf die externen Deklarationen wird in einer baumartigen Stapelstruktur mit einer Anordnung erreicht, bei welcher das zuletzt eingegebene zuerst ausgegeben wird und welche eine schnellere Suchreihenfolge für externe Deklarationen gestattet.

DE 43 13 958 A 1

Die Erfindung betrifft eine Einrichtung zum Steuern der Darstellung einer Vorlage nach dem Oberbegriff des Anspruchs 1 oder 2 und betrifft ein Verfahren zum Steuern der Darstellung einer Vorlage nach dem Oberbegriff des Anspruchs 10. Ferner betrifft die Erfindung generell ein rechnergesteuertes Drucken von Vorlagen mit Text- und graphischen Komponenten, und betrifft insbesondere ein Verfahren und ein System für eine Implikation (inclusion) externer Dateien, die in einer strukturierten Seitenbeschreibungssprache dargestellt sind, in einen Datenstrom, der gedruckt oder dargestellt werden soll. Dies ist insbesondere für eine automatische Verschiebung zwischen verschiedenen Daten und Druckformaten verwendbar, welche sich für eine Vielzahl von Darstellungseinrichtungen und/ oder für die Datenübertragung zu anderen Einrichtungen in einem Übertragungssystem eignen.

In diesem Zusammenhang schließt die Darstellung einer Vorlage auf einer Seite das Drucken auf einem festen Medium, wie Papier, einem transparenten Material u.ä., ein Darstellen einer Seite auf einer visuellen Anzeige oder ein Übertragen der Vorlage in einer darzustellenden oder zu speichernden Form an eine andere Einrichtung mit ein. Die vorliegende Anmeldung hat auch Bezug zu einer anhängigen, am 17. Oktober 1991 eingereichten US-Anmeldung S.N. 07/778 578, auf deren Inhalt hiermit Bezug genommen wird. Die vorliegende Anmeldung hat ferner Bezug zu einer gerade eingereichten US-Patentanmeldung mit dem Titel "Verfahren und Einrichtung zum Verwalten eines Bildes und Seitensatzes für eine Vorlagenverarbeitung", auf welche hiermit ebenfalls Bezug genommen wird.

Durch die Entwicklung von Laserdruckern in den frühen Siebziger-Jahren wurde eine Möglichkeit für hochqualitatives und preiswertes Drucken von Vorlagen geschaffen, welche nicht nur Zeichentext, sondern auch übliches grafisches Material enthalten. Bei dem grundlegenden Verfahren, um Befehle von dem Rechner zu dem Drucker zu steuern, wurden die sogenannten "Escape-Folge"-Befehle verwendet, welche den Befehlen ähnlich sind, die in der DIABLO®-Befehlsfolge verwendet sind. Diese Befehlsarten wurden durch typische Zeichendaten unterschieden, wobei jeder Befehl durch ein spezielles Byte, im allgemeinen ein Escape-Zeichen (ASCII 27) eingeleitet wird. Diese Methodologie arbeitete in annehmbarer Weise mit Typenrad- oder Punktmatrix-Druckern, war allerdings nicht zum Drucken von Vorlagen geeignet, welche sowohl Text- als auch grafische Materialien enthalten, da keine Vorkehrungen für große Datenmengen vorgesehen waren, welche eine grafische Darstellung erfordern kann. Ebenso konnten keine externen Dateien schnell in einen Datenstrom einbezogen werden.

Als Antwort auf die Beschränkungen im Falle der Escape-Folge-Befehle wurden verschiedene Arten von "Seitenbeschreibungssprachen" (PDL) entwickelt, um Laserdrucker oder andere Arten von Seitendruckern zu steuern. Eine rückwärts wirkende Kompatibilität war bezüglich der meisten dieser Laserdrucker vorgesehen, so daß sie im allgemeinen Escape-Folge-Befehle annehmen konnten. Beispiele von solchen Seitenbeschreibungssprachen sind das PostScript®-System von Adobe Systems Incorporated und das InterPress®-System der Xerox®-Corporation. Es sind auch noch weitere gesetzlich geschützte Seitenbeschreibungssprachen bekannt.

Die herkömmlichen Seitenbeschreibungssprachen

weisen verschiedene Verbesserungen bezüglich der vorher bekannten Standard-Escape-Folgen auf, wie ein Vorsehen von Ressourcen-Deklarationen, Kontext-Deklarationen, Wörterbüchern, das Verwenden von Speicherstapeln u.ä. Diese Sprachen ließen in einigen Fällen auch dynamische Änderungen in dem Druckerzustand zu, so beispielsweise die Möglichkeit, Schriftarten oder grafische Bilder zu den Ressourcen hinzuzufügen, die in dem Drucker verfügbar sein können. Einige dieser Merkmale sind in solchen im allgemeinen verfügbaren Referenzarbeiten dokumentiert, wie in "PostScript Language Reference Manual" von Adobe System Incorporated und das "PostScript Language Programm Design", beide von Addison-Wesley Publishing Company. Andere Seitenbeschreibungssprachen sind ebenfalls in verschiedenen technischen und Referenzbüchern beschrieben, wie in "InterPress, The Source Book" von Harrington et. al. (Simon and Schuster, Inc., 1988).

Eine Standard-Seitenbeschreibungssprache ist vorgeschlagen worden und zu einer internationalen Norm entwickelt worden. Der Vorschlag, bei welchem der Erfinder der vorliegenden Anmeldung mitgewirkt hat, liegt zum gegenwärtigen Zeitpunkt in Form eines Entwurfes einer Abteilung der ISO vor. Der Entwurf ist bekannt als ISO/IEC DIS 10180 und wird bezeichnet als "Informationsverarbeitung-Textübermittlung-Standard-Seitenbeschreibungssprache". Der aktuelle Entwurf datiert von 1991.

Viele der bekannten Arten von Seitenbeschreibungssprachen weisen verschiedene Mängel auf, einschließlich der Tatsache, daß sie im allgemeinen auf eine einzelne Seite beschränkt sind und im allgemeinen nicht eine voll strukturierte Sprache verwenden. Beispielsweise ist einer der Nachteile der PostScript®-Sprache der, daß die Seitenbeschreibung für eine spezielle Vorlage eine neue Definition enthalten kann, wie beispielsweise eine Ressourcen-Definition (beispielsweise eine zusätzliche Schriftart) oder eine neue Wörterbuch-Definition, welche irgendwo in der Vorlage verwendet werden kann. Als Folge hiervon muß der gesamte Inhalt der Vorlage verarbeitet werden, um zu bestimmen, ob ein spezieller Drucker die Ressourcen hat, die notwendig sind, um sie zu drucken. Andererseits kann, wenn diese "Vorverarbeitung" nicht durchgeführt wird, das Drucken einer Vorlage an irgendeiner Stelle während des Druckprozesses sogar kurz vor dem Ende scheitern, da der Drucker nicht in der Lage ist, Befehle der Vorlagenseitenbeschreibung zu befolgen.

Bei den herkömmlichen Systemen, welche die PostScript-Sprache verwenden, ergeben sich weitere Schwierigkeiten, das, um eine vorgegebene Seite einer Vorlage zu drucken, es allgemein notwendig ist, am Anfang jeder Seite die gesamte PDL-Beschreibung aller vorhergehenden Seiten der Vorlage zu lesen, um den Zustand der Vorlagen-Seiteneinstell-Parameter (d. h. Ressourcen-Deklarationen, Wörterbuch-Definitionen u.ä.) zu bestimmen. Mit anderen Worten, ein Drucksteuer- oder Druck-Treiberprogramm muß die gesamte PDL-Beschreibung einer Vorlage lesen, um die Wirkung jedes Seiteneinstellbefehls zwischen dem Anfang der Vorlage und der bestimmten Seite in Betracht zu ziehen. Obwohl dieser Seiteneinstell-Abtastvorgang verhältnismäßig unkompliziert ist, erfordert er eine beträchtliche Prozessorzeit, welche besser von dem Drucker benutzt werden kann.

Außerdem gibt es keine Syntax oder Semantik, welche in der PostScript®-Sprache definiert ist, um die Implikation externer Objekte zu behandeln, noch gibt es

irgendeine einfache Methode, dies Merkmal hinzuzufügen.

Eine der Schwierigkeiten bei den bekannten Inter-Press®-System von Xerox® ist, daß dessen Struktur es nicht gestattet, entweder "Klartext" oder binäre Datenströme abwechselnd zu benutzen. Um externe Objekte (entities) eines Vorlagen-Datenstrom hinzuzufügen, kann ein "globaler" Änderungstyp in Vorlagen vorgesehen sein, wobei eine Bibliothek von "Stil"-Standardisierungen bequem geschaffen werden kann, so daß normierte oder vorgeschriebene Stile für Arbeitsgruppen, Geschäftsbereiche oder unternehmensweite Vorgänge definiert werden können.

Gemäß der Erfindung soll daher ein Verfahren und eine Einrichtung für die Implikation (inclusion) von externen Deklarationen in einer schnellen Weise geschaffen werden.

Die Implikation von externen Objekten oder Deklarationen ist wichtig, um eine höhere Flexibilität für eine Vorlagen-Beschreibungssprache oder den Datenstrom zu schaffen, welcher eine Vorlage beschreibt, wenn sie zu drucken ist. Insbesondere wenn Änderungen in der Art der Vorlagenvorbereitung vorkommen oder wenn neue Einrichtungen für einzelne Druckvorrichtungen u.ä. wie beispielsweise Endverarbeitungseinheiten, Hefter oder andere Vorrichtungsarten verfügbar werden, welche nicht erwartet wurden, erhöht die Implikation externer Deklarationen die Flexibilität. Die Implikation externer Objekte in der vorliegenden Erfindung erlaubt es, die Druckeinrichtung oder die Anzeigeeinheit hinzuzufügen diese neuen Merkmale zu verwenden und die Verwendung dieser Merkmale in einem Vorlagen-Datenstrom einzufügen, ohne daß die Sprache wieder definiert werden muß. Hierdurch wird die Flexibilität des Systems erhöht und erlaubt eine weit größere Benutzung von Ressourcen ohne das voraussehbare Veraltern wie bei den meisten herkömmlichen Systemen.

Ferner soll gemäß der Erfindung eine Einrichtung geschaffen werden, bei welcher Vorlagen-Datenströme mit Hilfe von externen Datendeklarationen in einer Weise modifiziert werden können, welche effizient verarbeitet werden kann. Ferner soll gemäß der Erfindung eine Einrichtung und eine Methodik zum Ändern von Datenströmen geschaffen werden, welche für das Drucken einer Vorlage mit externen Datendeklarationen in einer strukturierten Lösung verwendet werden oder verwendet werden sollen. Ferner soll noch eine Einrichtung und eine Methode geschaffen werden, um die Leistungsfähigkeit eines Druckers mit Hilfe externer Deklarationen in einer strukturierten Umgebung hinzuzufügen.

Diese und andere Ziele sind gemäß der Erfindung dadurch erreicht, daß eine Einrichtung zum Steuern der Darstellung von Vorlagen geschaffen ist, die in einer vorlagenbeschreibenden Sprachform dargestellt sind. Externe Objekte oder Dinge, wie Graphiken oder andere Datenströme (welche andere vollständige Vorlagen sein können) können zu einem Vorlagen-Datenstrom hinzugefügt werden, so daß die Definitions- und beschreibenden Befehle, welche den externen Dingen oder Objekten zugeordnet sind, ohne weiteres verarbeitet werden. Diese externen Vorlagen oder Objekte sind ebenfalls strukturiert. Gemäß der Erfindung wird dies dadurch erreicht, daß jeder Vorlagen-Datenstrom null oder mehr Prologabschnitt schafft, welche Definitionen und erläuternde Befehle enthalten und mit Inhaltsteilen versehen werden, welche spezifische Berechtigungsmarken (token) oder Befehle zum Definieren spezifi-

scher Bilder enthalten. Die Einleitungsabschnitt und Inhaltsteile werden in einer baumförmig verknüpften Hierarchie angeordnet. Diese Hierarchie berücksichtigt externe Deklarationen, wobei die am kürzesten eingegebene, externe Deklaration bezüglich früher eingegebener Deklarationen höhere Positionen in der Hierarchie besetzt.

Die baumförmig verknüpfte Hierarchie schafft einen wichtigen Vorteil in der vorliegenden Erfindung, da ein externes Ding oder irgendein anderer Teil der Vorlage verarbeitet oder gedruckt werden kann, ohne irgendeinen anderen Teil der Vorlage verarbeiten zu müssen. Nur strukturelle Definitionen in dem hierarchischen Baum, welche oben ein ausgewählter Teil einer beispielsweise Seite drei einer zwanzigseitigen Vorlage verarbeitet werden muß, müssen nur Seite drei und die begleitenden Deklarationen, welche auf einem Pegel Seite 3 oder darüber in dem hierarchischen Bau liegen, verarbeitet werden. Dies erhöht die Effizienz und erleichtert auch die Bestimmung des Ressourcen-Pegels, welcher bei einer Vorlage vor dem Beginn des tatsächlichen Druckens der Vorlage benötigt wird. Bezüglich externer Deklarationen werden hierdurch beträchtliche Verbesserungen hinsichtlich der Verarbeitungsgeschwindigkeit erreicht und es wird auch nur die Verarbeitung dieser externen Deklarationen gefordert, welche notwendig sind.

Nachstehend wird die Erfindung anhand von bevorzugten Ausführungsformen unter Bezugnahme auf die anliegenden Zeichnungen im einzelnen erläutert. Es zeigen:

Fig. 1 eine Darstellung einer Stapelanordnung, welche eine Spur eines Bildes und eines Seitensatzes einer Vorlage hält, welche in der strukturierten Standard-Seitenbeschreibungssprachenform der vorliegenden Erfindung dargestellt ist;

Fig. 2 eine Darstellung von Schritten in der Zeigermethode, um externe Dinge zu der in Fig. 1 dargestellten Datenstruktur hinzuzufügen;

Fig. 3 eine Darstellung der Verarbeitung eines zweiten Bildes oder Seitensatzes;

Fig. 4 eine schematische Darstellung der Zusammenschaltung zwischen zwei Prozessoren, welche die externe Deklaration und eine externe Identifizierung in einer SPDL-Vorlage gemäß einer Anwendung dieser Erfindung verarbeiten mit einem lexikalischen Analysator und Parser;

Fig. 5 ein Flußdiagramm von Verarbeitungsschritten eines externen Deklarationsprozessors gemäß der Erfindung;

Fig. 6 ein Flußdiagramm von Verarbeitungsschritten eines externen Identifizierprozessors gemäß der Erfindung;

Fig. 7 eine Darstellung der Wirkungen, um das externe Objekt in einem SPDL-Vorlagenstrom zu ersetzen, und

Fig. 8 ein Blockdiagramm eines Beispiels einer SPDL-Struktur gemäß der Erfindung.

In Fig. 1 sind der Bild- und Seitenstapel gemäß der Erfindung dargestellt. Es wird eine Stapelanordnung verwendet, in welcher ein Zeiger zu einem "Prolog" bzw. einer Einleitung 100, die Information schafft, die notwendig ist, um die externe Deklaration zu einer vorgegebenen Vorlagenbeschreibung hinzuzufügen. Der vorgegebene Seitensatz-Pegel ist in einem Feld 1002 definiert. Der vorgegebene Bildpegel, welcher einem vorgegebenem Prolog zugeordnet ist, ist im Feld 1004 gesetzt. Eine Reihe von Zeigern 1006, 1008, 1010, 1012, 1014,

1016, 1018 und 1020 werden als Zeiger zu verschiedenen Feldern bzw. Elementen verwendet, wie der externen Deklaration, einer informellen Deklaration, ein Zeiger zu einer Ressourcen-Definition, ein Zeiger einer Ressourcen-Deklaration, ein Zeiger zu einer Context-Deklaration, ein Zeiger zu einem Wörterbuch-Generator und ein Zeiger zu einer Einstellprozedur bzw. ein Zeiger zu dem Gerätezustand. Eine Vorlagenbeschreibung kommt von dem strukturierten Teil eines Vorlagen-Datenstroms. Eine Suchspur durch den Vorlagen-Datenstrom wird auf der Basis der zusätzlichen "Unterzeiger" 1006 bis 1020 eingestellt, welche festgelegt werden. Die Suchspur geht von dem ersten Zeiger in dem Anfangsprolog zu einer externen Deklaration 1006, der informativen Deklarationen 1008, der Ressourcen-Definition 1010, der Ressourcen-Deklaration 1012, der Context-Deklaration 1014, dem Wörterbuch-Generator 1016, der Einstellprozedur 1018 und zu dem Gerätezustand 1020 weiter.

In Fig. 8 ist die Anordnung einer herkömmlichen SPDL-Vorlage dargestellt, wie sie durch die Erfindung verwendet oder verarbeitet würde. An dieser Stelle sollte beachtet werden, daß die Grundstruktur des Vorlagen-Datenstroms so ist, wie in der vorerwähnten, vorgeschlagenen 150-Norm festgelegt ist. Jede Vorlage enthält eine Vorlagenbeschreibung, welche sowohl Struktur- als auch Inhaltelemente hat. Die Vorlagenstruktur und ihre diesbezüglichen Elemente sind von dem Inhalt der Vorlage unabhängig und können unabhängig von dem Inhalt einer Vorlage verarbeitet werden. Der Inhalt liegt in Form einer Seitenbeschreibungssprache vor. Seine Verarbeitung ist unabhängig von der Vorlagenstruktur, deren Elemente und dem Kontext, in welchem sie vorkommt.

Wenn eine Strukturverarbeitung der Darstellungszweck ist, bildet die Strukturverarbeitung den Interpretationskontext jedes Elements eines Vorlageninhaltes. Wenn jedoch die Strukturverarbeitung für andere Zwecke als eine Vorlagendarstellung erfolgt, ist es üblicherweise nicht notwendig, eine Inhaltverarbeitung durchzuführen. Der Darstellungsprozeß kann auch verfügbar sein für verschiedene Quellen zusätzlicher Daten, wie Schriftarten, die in Teilen von externen SPDL-Vorlagen gespeichert sind, auf welche durch die SPDL-Vorlage Bezug genommen werden kann, welche zu verarbeiten ist. Auf die Summe aller solcher Daten wird als die Umgebung des Darstellungsprozesses Bezug genommen.

Die SPDL-Vorlagenstruktur ist das Ergebnis der Teilung und der wiederholten Unterteilung einer SPDL-Vorlage in immer kleinere Teile. Jeder der sich ergebenden Teile wird als Strukturelement bezeichnet. Die SPDL-Vorlagenstruktur ist in ihrer Art hierarchisch. Die höchste Ebene ist eine Vorlage 80. Eine Vorlage kann untergeordnete Strukturelemente enthalten, welche als Seitensatz bzw. Seitensatz 82 oder Bildstrukturelemente bezeichnet werden.

Ein Seitensatz 82 kann seinerseits wiederum untergeordnete Seitensätze und/oder Bilder aufweisen. Jedes Bild enthält eine Beschreibung der Vorlage. Der höchste Bildpegel in der Bildhierarchie ist eine Seite, welche in einem einzigen Fall des Darstellungsmediums (d. h. beispielsweise einer einzigen Seite) dargestellt worden ist.

Ein Bild kann untergeordnete Strukturelemente, die als Bilder 88 bis 96 bezeichnet sind, und/oder Tokenfolge-Strukturelemente enthalten. Eine Tokenfolge ist ein Strukturelement, welches Vorlageninhalt enthält. Eine Vorlage ist einfach ein Seitensatz bzw. -set oder ein Bild,

welches auf dem höchsten Niveau in der Strukturhierarchie angeordnet ist. Dementsprechend ist eine Seite einfach ein Bild, welches auf dem höchsten Pegel in der Hierarchie liegt.

In diesem Zusammenhang weisen der Term-Zeiger und der Unterzeiger auf das herkömmliche Datenstrukturkonzept hin, bei welchem Information als eine Dateneingabe, nicht als die Daten selbst vorgesehen sind. Dies bedeutet, die Zeiger sind Daten, die auf eine vorgegebene Speicherzelle oder auf einen anderen Zeiger hinweisen, welcher auf die Speicherstelle der tatsächlichen Daten oder auf die tatsächlichen Daten selbst hinweist. Natürlich sollte diese Definition in ihrer Art als rekursiv verstanden werden. Ferner sollte zum Verständnis dieser Erläuterung angenommen werden, daß Ressourcen, Wörterbuch- und externe Definitionen in ihrer Funktion ähnlich den Ressourcen- und Wörterbuchdefinitionen sind, welche in der PostScript®-Seitenbeschreibungssprache zu finden sind. Ressourcen-Deklarationen und -Definitionen sind im allgemeinen auf solche Begriffe gerichtet, wie Schriftarten, Filter, Füllmuster, Farben, Glyphen u.ä. und können verfügbar sein, durch Aufrufen einer Tokeneinheit in dem Datenstrom, welcher durch den Zeiger oder eine Ressourcen Einheit geschaffen ist. Ressourcen-Deklarationen binden einen Namen an eine ganz bestimmte Ressource, während Ressourcen-Definitionen die exakte Beschaffenheit jeder Ressource liefern. Wörterbücher werden verwendet, um Schlüsselwerte in vorgegebenen Listen von Merkmalen (Token) oder andere Werte zu übersetzen und können ähnlich einer Makro-Definition betrachtet werden. Eine externe Definition verweist auf eine Untergruppe von Strukturelementen, welche außerhalb einer vorgegebenen Vorlage liegen und keine echte eins-zu-eins-Beziehung zu irgendetwas in der PostScript®-Sprache haben.

Ein externes Objekt kann beispielsweise eine graphische Darstellung oder ein Bild in Form eines Tokenfolge-Strukturelements sein, welches getrennt von der Vorlage ist, aber welches automatisch während der Verarbeitung der Vorlage aufgenommen wird. Es kann beispielsweise ein Bildstrukturelement oder eine andere Art eines graphischen Tokenfolge-Strukturelements sein, welches beispielsweise als ein Kopf- oder Fußteil auf jeder Seite enthalten ist. Es kann auch ein anderer Strukturelement-Datenstrom sein, und daher gestattet die Definition eine Rekursion der strukturierten Elemente. Da sich jedoch externe Deklarationen auf der Basis der Verarbeitung, welche stattfindet, ändern können, muß eine Verarbeitung in einer vorgegebenen Druck- oder Anzeigeeinrichtung stattfinden, so daß das externe Objekt zur Darstellung für den Benutzer vorbereitet werden kann. Es ist wichtig, externe Objekte "im Flug" hinzuzufügen oder ändern zu können. Dies wird durch die in Fig. 2 dargestellte Zeigerstruktur erreicht.

Insbesondere führt die in Fig. 2 dargestellte Zeigermethode einen Zeiger bei der externen Deklaration durch, was in der in Fig. 1 dargestellten Prolog-Datenstruktur dargestellt ist.

Anfangs zeigt dieser Zeiger auf eine "null"; bevor irgendeine externe Deklaration angetroffen wird, wird der Speicher, welcher für eine externe Deklarations-Datenstruktur notwendig ist, der Druck- oder Anzeigeeinrichtung zugeteilt. Dieser Zeiger wird dann für einen Null-Wert an der Adresse des zugeteilten Speichers für die externe Deklaration-Datenstruktur geändert. Dieses Element, das als nächstes in der externen Datenstruktur bezeichnet wird, zeigt dann auf null, wenn nur eine einzi-

ge externe Deklarations-Datenstruktur vorhanden ist. Andere Objekte in dem externen Datenstrukturen werden verarbeitet, wenn sie in einer ähnlichen Weise angetroffen werden, wobei der frühere Zeiger geändert wird, um auf den neu zugeteilten Bereich zu zeigen, und wobei der Zeiger in dem neu zugeteilten Bereich entsprechend eingestellt wird, um auf das vorher erste, externe Objekt zu zeigen, wie in Fig. 2 dargestellt ist.

Diese externen Deklarations-Datenstrukturen werden in der in Fig. 4 und 5 wiedergegebenen Weise verarbeitet, was nachstehend noch erläutert wird. Wenn andere externe Deklarationen auf einer Vorlage angetroffen werden, werden sie vor der alten externen Deklaration plziert, wobei die Zeiger entsprechend geändert werden. Mit anderen Worten, wenn sie verarbeitet werden, werden neue externe Deklarationen einer höheren hierarchischen Position als vorher verarbeitete, externe Deklarationen zugeordnet. Folglich befindet sich die neueste externe Deklaration als erste in einer Suchanweisung bezüglich der "Typenpassungs"-Informationshaltespur, ob das externe Objekt korrekt definiert ist oder nicht.

Fig. 3 zeigt den Zeiger, wenn ein zusätzlicher Eintrag des BILD/SEITENSATZ-Stapels gemacht wird. Der zusätzliche Eintrag zeigt auf eine neue PROLOG-DATENSTRUKTUR. Die externe Deklaration dieser neuen Datenstruktur zeigt auf dieselbe Stelle wie die Stelle, auf welche durch die externe Deklaration einer PROLOG-DATENSTRUKTUR gezeigt wird, auf welche durch den Eintrag unten im BILD/SEITENSATZ-Stapel gezeigt worden ist.

Das Verarbeiten des zweiten Bildes oder Seitensatzes kann ausgeführt werden. Wie in Fig. 3 dargestellt, zeigt ein Zeiger auf eine Adresse einer externen Deklaration einer Prolog-Datenstruktur. Dieser Zeiger kennzeichnet und lokalisiert die Adresse in der externen Deklaration einer Prolog-Datenstruktur, auf welche durch ein nachfolgendes Objekt in dem Bild/ Seitensatz-Stapel gezeigt wird, der auf dieses externe Objekt verweist. Durch Durchführen dieser Bild-Seitensatz-Anordnung mit Hilfe eines Stapels werden die Datenstrukturen geordnet. Wie in Fig. 3 dargestellt, zeigt, selbst wenn eine der niedrigeren Prolog-Datenstrukturen Mehrfach-Zeiger hat, (der Zeiger zu der externen Deklaration 70 beispielsweise mehrfache externe Deklarationen hat), der Zeiger auf die neueste externe Deklaration, obwohl frühere externe Deklarationen für die externe Deklaration dargelegt worden sind. Dies kommt wegen der Anforderung vor, daß externe Deklarationen, welche erst vor kurzem hinzugefügt worden sind, vor früher eingegebenen externen Deklarationen angeordnet werden. Folglich wird die letzte externe Deklaration immer zuerst in der Suchreihenfolge gesucht, wodurch die Prozessorzeit verkürzt wird, die zum Suchen zusätzlicher externer Deklarationen notwendig ist.

Wenn der Pegel oder Bereich in der Hierarchie einer externen Deklaration, welche zu einem geordneten Datenstrom hinzuzufügen ist, in den geordneten Datenstrom eingesetzt wird, kann ein externes Strukturelement durch Verweis lexikalisch enthalten sein, wenn es insgesamt in dem Vorlagen-Datenstrom eingebracht wird, indem der externe Kennzeichner in die Struktur an irgendeiner Stelle eingebracht wird, wobei ein entsprechendes externes Element akzeptabel sein würde. Diese externe Deklaration wird hinzugefügt und würde durch den Wert der Strukturkennzeichner gekennzeichnet werden. Der Strukturprozessor, welcher nachstehend noch erläutert wird, löst irgendwelche Referenzen zu

externen Kennzeichnern dadurch auf, daß das externe Strukturelement für den externen Kennzeichner ersetzt wird.

Jede externe Deklaration bindet einen externen Kennzeichner an ein externes Strukturelement. Eine externe Deklaration, wie sie durch die Zeichen-ISO-Norm definiert ist, ist eine zusammengesetzte Struktur, welche mehrere unmittelbare untergeordnete Elemente hat, wie ein externes Kennzeichner-Strukturelement, ein Strukturtyp-Kennzeichner-Struktur-Element und ein Strukturkennzeichner-Strukturelement. Die Strukturelemente kommen in einer ganz bestimmten Reihenfolge vor. Irgendein externer Kennzeichner, welche in einer externen Deklaration deklariert worden ist, kann irgendwo in der Vorlagenstruktur verwendet werden, in welcher ein Strukturelement der Art, welche durch die externe Deklaration genau bestimmt worden ist, ein gültiger Typ sein würde.

Die Wirkung des Vorkommens eines externen Kennzeichners in einem Strukturelement, welches in dem Bereich einer externen Deklaration liegt, in welcher der externe Kennzeichner an das externe Strukturelement gebunden worden ist, bewirkt, daß diese externe Deklaration als ein Ersatz für den ausgewechselten externen Kennzeichner wirkt. Der Strukturprozessor, welcher nachstehend noch beschrieben wird, löst irgendwelche Referenzen zu den externen Deklarationen auf, indem das "korrekte" externe Strukturelement für den externen Kennzeichner gesetzt wird, der auf dem entsprechenden Pegel in der Hierarchie ist. Das externe Strukturelement wird dann von dem Strukturprozessor verarbeitet, obwohl es ein Teil eines ursprünglichen SPDL-Vorlagen-Datenstroms gewesen ist. Auf diese Weise kann, sofern die endgültige Verarbeitung erfolgt, sobald ein externes Objekt durch die externe Deklaration "eingebunden" worden ist, es in derselben Weise verwendet werden, wie wenn es ein Teil der ursprünglichen Vorlage wäre, ohne daß irgendeine zusätzliche Verarbeitung notwendig ist.

Wie in Fig. 4 dargestellt, erfordert ein lexikalischer Analysierer und Parser 40, welcher in der Erfindung verwendet ist, logisch getrennte Mehrfachprozessoren. Im Rahmen einer Darstellung der Erfindung sind jedoch nur der externe Deklarationsprozessor 42 und der externe Kennzeichnerprozessor 44 von Bedeutung. Wie anhand des Beispiels erwähnt, das in Fig. 8 dargestellt ist, wird der externe Kennzeichner während der Deklarationsphase verwendet und kennzeichnet die Stelle, an welcher die Implikation (inclusion) einer externen Referenz angeordnet ist. Die zwei Prozessoren können entweder in einzelnen Schaltungen oder als Teil eines auswählbaren SPDL-Prozessors ausgeführt sein. Die Prozessoren 1 legen auf der Basis eines eingegebenen Vorlagen-Datenstroms die Parameter fest, welche durch diese zu Prozessoren in der nächsten Stufe bei der Darstellungsverarbeitungen durchgelassen werden. Der externe Deklarationsprozessor wird im einzelnen anhand von Fig. 5 erläutert, während der externe Kennzeichnerprozessor im einzelnen anhand von Fig. 6 erläutert wird.

In Fig. 5 sind die Verarbeitungsschritte des externen Deklarationsprozessors gemäß der Erfindung dargestellt. Die externe Verarbeitung, welche in Fig. 5 stattfindet, ist wichtig, da sie ein Mittel schafft, durch welches externe Deklarationen in den eingegebenen Vorlagen-Datenstrom in einer Weise "eingebunden" werden können, welche die Implikation von externen Deklarationen in effizienter Weise berücksichtigt. Die generelle Ar-

beitsweise des externen Deklarationsprozessors wird nunmehr anhand von Fig. 5 beschrieben. Beim Schritt 102 ordnet der externe Deklarationsprozessor einen Speicher für eine externe Deklarations-Datenstruktur zu und ändert den Zeiger in der korrekten und neuen externen Deklaration 102. Beim Schritt 104 werden der Seitensatz-(page set)Pegel, der Bildpegel, die Werte des externen Kennzeichners, der Strukturtyp und die Quelleninformation gesetzt. Beim Schritt 106 wird die Informationsdatei, welche das Quellenmaterial für die neue externe Definition enthält, geöffnet, und beim Schritt 108 wird diese Datei gelesen, bis das "erste Struktur-Hinweissymbol (tag)" getroffen wird. Wenn das Ende der Datei-Hinweissymbols das erste Hinweissymbol im Schritt 110 ist, wird die Typenanpassung beim Schritt 116 auf "nein" gesetzt. Wenn beim Schritt 110 das erste Hinweissymbol, welches in dieser geöffneten Datei angetroffen wird, nicht ein Ende eines Datei-Hinweissymbols ist, dann wird beim Schritt 112 eine Prüfung des Hinweissymboltyps vorgenommen, um zu bestimmen, ob es zu dem Strukturtyp paßt. Wenn nicht, wird auch der Schritt 116 durchgeführt. Wenn jedoch beim Schritt 112 das Typengegenstück paßt, wird der Schritt 114 durchgeführt, und das Typen-Gegenstück wird "ja" gesetzt. An diesem Punkt geht dann die Verarbeitung beim Schritt 118 weiter, um zu zeigen, daß der nächste Wert gesetzt wird, der an diesem Punkt oder an der externen Deklaration in der laufenden Prolog-Datenstruktur gleich ist. Dadurch kann dann die neue Deklaration zuerst in der Hierarchie angeordnet werden, und der Zeiger zu der externen Deklaration der aktuellen Prolog-Datenstruktur ist gleich dem laufenden ptr-Wert (ptr-current) beim Schritt 120 gesetzt. Beim Schritt 122 wird dann die externe Deklaration geschlossen, und es wird ein Rücksprung durchgeführt, wodurch die Verarbeitung fortgesetzt werden kann. Mit anderen Worten, der externe Kennzeichner kommt von dem <EXTID>-Hinweissymbol her, wobei der Strukturtyp von dem Struktur-Attribut kommt, und die Quelleninformation von dem <STRUCTID>-Hinweissymbol, wenn die Beispiel-Information zu verarbeiten war.

Der in Fig. 4 dargestellte, externe Kennzeichnerprozessor arbeitet so, wie in Fig. 6 dargestellt ist. Pauschal gesagt, die laufende Eingangsdatei und der Eingangsdateistapel stehen diesem Prozessor zur Verfügung, und eine Verarbeitung beginnt beim Schritt 130, wobei die Prolog-Datenstruktur durch die Oberseite des Bild/Seitensatz-Stapels angezeigt wird. Beim Schritt 132 wird bestimmt, ob der Zeiger zu der externen Deklaration null ist. Wenn dem so ist, kommt ein Fehler vor, und das Fehler-Flag wird beim Schritt 134 gesetzt. Wenn jedoch der Zeiger zu der externen Deklaration nicht null ist, wird beim Schritt 136 der externe Kennzeichner geprüft, um zu bestimmen, ob er gleich Ex\_id ist. Wenn er dies nicht ist, wird beim Schritt 138 der nächste Zeiger erhalten, und die Verarbeitung kehrt auf den Schritt 132 zurück, um zu bestimmen, ob die externe Deklaration null ist. Wenn sie andererseits beim Schritt 136 gleich ist, geht die Verarbeitung auf den Schritt 140 über, um zu bestimmen, ob das Gegenstück vorhanden ist. Wenn nicht, wird das Fehlerflag beim Schritt 134 gesetzt, und es wird eine Rückkehr durchgeführt. Wenn es paßt, dann wird der Strukturtyp mit dem (den) gültigen Strukturtyp(en) beim Schritt 142 verglichen, und wenn nicht festgestellt wird, daß er korrekt ist, wird wieder das Fehlerflag gesetzt, und eine Rückkehr durchgeführt. Wenn beim Schritt 142 der Strukturtyp korrekt ist, dann wird beim Schritt 144 die externe Datei mit Hilfe der

Quelleninformation geöffnet, welche in der Daten-Variablen enthalten ist, auf welche durch den Zeiger gezeigt wird. Beim Schritt 146 wird die aktuelle Eingabedatei gesichert, indem ein Dateizähler in dem Dateistapel des Prozessors eingespeichert wird, wobei diese seit kurzem offene Datei zu der aktuellen Eingabedatei beim Schritt 148 gemacht wird.

Durch die vorstehend beschriebene Verarbeitung ist eine größere Flexibilität gegenüber herkömmlichen Systemen geschaffen, da die externe Datei eingeschlossen werden kann und eine neue Ressource gemacht worden ist, und es ist die Implikation (Einbeziehung) eines externen Objekts in den Hauptteil der ursprünglichen Vorlage gestattet. Ein Beispiel hierfür ist in Fig. 7 dargestellt, bei welchem das gesamte externe Objekt (entity) 72 in eine ursprüngliche Vorlage 70 eingesetzt wird, wodurch eine Implikation einer externen Deklaration in einen individuellen Seitenbeschreibungssprachen-Prozeß gestattet ist. Ein wichtiges Ergebnis dieser Einsetzung/Implikation ist, daß der "Name" oder die Identifikation der externen Objekte dasselbe (d.h. "Kopf-Datei") sein kann; jedoch basierend auf dem Kontext, wobei in dem Vorlagendatenstrom der Ruf zu dem externen Objekt gemacht wird, wird das "korrekte" externe Objekt gefunden und von dem Strukturprozessor angewendet. Infolge der in Fig. 3 dargestellten Zeigeranordnung kommt es immer zu einer korrekten Anpassung. Ferner gestattet dies einem externem Dokument, neue Aufrufe zu noch einem anderen externen Dokument zu erhalten.

Ein Merkmal, das durch diese Implikation (Einbeziehung) ermöglicht wird, besteht darin, daß beispielsweise durch Änderung eines externen Dokuments ein neues Firmenlogo auf alle Vorlagen eingebracht werden kann, welche vorher das "alte" Logo erforderten, wenn der Abruf durch externe Deklaration erfolgte. Ferner ist weitaus leichter eine stärkere Nutzung von gemeinsam benutzten Ressourcen möglich, wie von fern- bzw. programm-ladbaren Graphiken. Systemweite Normänderungen oder abteilungsweite Änderungen sind leichter durchzuführen und aufgrund dieses Merkmals leichter zu steuern. Durch die Suchreihenfolge, welche wegen der Zeigeranordnung in Fig. 3 entsteht, können globale Änderungen durchgeführt werden, wobei eine korrekte Anbindung gewährleistet ist.

#### Patentansprüche

1. Einrichtung zum Steuern der Darstellung einer Vorlage mittels einer Darstellungsvorrichtung, gekennzeichnet durch eine Eingabeeinrichtung zum Aufnehmen eines eingegebenen Vorlagendatenstroms, welcher die Vorlage festlegt, wobei der Datenstrom sich aus einem Bild mit null oder mehr externen Verweisen oder einem Seitensatz mit null oder mehr externen Verweisen mit einer hierarchisch geordneten Struktur zusammensetzt; wobei das Bild null oder mehr Prologe und null oder mehr Bildkörper aufweist, und der Seitensatz null oder mehr Prologe und null oder mehr Seitensatzkörper aufweist, wobei der Bildkörper null oder mehr Tokenfolgen und null oder mehr Bilder aufweist; wobei der Seitensatz null oder mehr Bilder und null oder mehr Seitensätze aufweist; eine Strukturverarbeitungseinrichtung, welche mit der Eingabeeinrichtung verbunden ist, um Seitensatz-Definitionen oder Bilddefinitionen festzule-

gen, um externe Verweise festzustellen, wobei die Strukturverarbeitungseinrichtung jeden Prolog in den vorherbestimmten Seitensatz- oder Bilddefinitionen verarbeitet, (Bild-) Körperteile zwischen festgestellten externen Verweisen in einer geöffneten Hierarchie ordnet und Ausgabevorlagen-Instruktionen zum Steuern der Darstellungseinrichtung erzeugt und abgibt, und  
 eine Abgabeeinrichtung, welche mit der Strukturverarbeitungseinrichtung verbunden ist, um die ausgegebenen Vorlageninstruktionen an die Darstellungsvorrichtung anzulegen.  
 2. Einrichtung zum Steuern der Darstellung einer Vorlage mittels einer Darstellungseinrichtung, gekennzeichnet durch  
 eine Eingabeeinrichtung zum Aufnehmen eines eingegebenen Vorlagendatenstroms, welcher die Vorlage festlegt, wobei sich der Datenstrom in einer geordneten hierarchischen Struktur mit null oder mehr Prologen, null oder mehr zugeordneten Inhaltteilen und null oder mehr Verweisen zu externen Vorlagendaten-Deklarationen befindet;  
 eine CPU-Einheit, welche mit der Eingabeeinrichtung verbunden ist, um den Datenstrom zu verarbeiten und um Abgabe-Vorlageninstruktionen auf der Basis des verarbeiteten Datenstroms zu erzeugen, mit  
 einer Einrichtung zum Ordnen des Adressierens zumindest eines Prologs und zugeordneter Inhaltsteile in einer geordneten Hierarchie und zum Ordnen des Adressierens der externen Vorlagendaten-Deklarationen in derselben geordneten Hierarchie, mit einer Einrichtung zum Verarbeiten zumindest eines Prologs und zugeordneter Inhaltsteile in der geordneten Hierarchie, um die Ausgangsvorlagen-Instruktionen zu erzeugen, und  
 eine Abgabeeinrichtung, die mit der CPU-Einheit verbunden ist, um die Ausgangsvorlagen-Instruktionen an die Darstellungsvorrichtung für die Darstellung der Vorlage anzulegen, welche durch die Ausgangsvorlagen-Instruktionen dargestellt ist.  
 3. Einrichtung nach Anspruch 2, dadurch gekennzeichnet, daß die Ordnungseinrichtung eine Stapelspeichereinrichtung aufweist, um Zeiger zu speichern, welche Speicherstellen zum Speichern von externen Vorlagendaten-Deklarationen ordnen, so daß die gespeicherten Speicher eine geordnete Auffindfolge schaffen.  
 4. Einrichtung nach Anspruch 2, gekennzeichnet durch  
 einen Inhaltsprozessor, der wirksam der CPU-Einheit zugeordnet ist, um den Inhalt des eingegebenen Vorlagendatenstroms zu verarbeiten;  
 einen Großspeicher, welcher mit der CPU-Einheit verbunden ist, wobei die externen Vorlagendaten-Deklarationen in dem Großspeicher gespeichert werden und aus dem Großspeicher unter der Steuerung des Inhaltsprozessors gelesen werden, wenn ein Verweis auf die externe Vorlagendaten-Deklaration durch die externen Deklarationen während einer Verarbeitung des Vorlagendatenstroms getroffen wird und der Zeiger auf die externen Deklarationsverweise von dem Großspeicher aus hinweist.  
 5. Einrichtung nach Anspruch 4, gekennzeichnet durch einen Randomspeicher, welcher mit der CPU-Einheit verbunden ist, um den Vorlagendatenstrom zu halten, und

eine Einrichtung, um von dem Großspeicher aus mit Hinweisen versehene, externe Vorlagendaten-Deklarationen, die in ihm und in dem Randomspeicher gespeichert sind, bei deren ersten Auftreffen durch den Inhaltsprozessor in den eingegebenen Vorlagendatenstrom zu übertragen, und  
 eine Einrichtung, um aus dem Randomspeicher die übertragenen externen Datendeklarationen bei dem ersten Auftreten eines Hinweises, daß der Speicherraum in dem Randomspeicher voll besetzt ist, oder bei dem Verarbeiten eines Inhaltselements in dem eingegebenen Vorlagendatenstroms zu entfernen, welcher denselben oder einen höheren Pegel in der geordneten hierarchischen Struktur des eingegebenen Vorlagendatenstroms hat.  
 6. Einrichtung nach Anspruch 2, dadurch gekennzeichnet, daß die Vorlagenstruktur-Verarbeitungseinrichtung einen Stapelspeicher aufweist, welcher mit der CPU-Einheit verbunden ist, um den Bereich der externen Vorlagendaten-Deklarationen zu steuern.  
 7. Einrichtung nach Anspruch 6, dadurch gekennzeichnet, daß der Stapelspeicher eine Zeigerstruktur benutzt, um eine Kennzeichnung der gespeicherten Gegenstände zu erhalten, so daß die letzte externe Vorlagendaten-Deklaration dem ersten Zeiger in dem Stapel zugeordnet wird.  
 8. Einrichtung nach Anspruch 7, gekennzeichnet durch eine Einrichtung, um die Zeiger in einem geteilten Speicher für die externen Vorlagendaten-Deklarationen in dem Stapelspeicher in einer verknüpften Listen-Anordnung zu speichern.  
 9. Einrichtung nach Anspruch 8, gekennzeichnet durch eine Sucheinrichtung zum Suchen des Stapelspeichers, so daß die Zeiger an einer externen Deklaration, welche auf einem niedrigeren Pegel in der Hierarchie des eingegebenen Vorlagendatenstroms liegt, vor irgendwelchen anderen Zeigern auf demselben oder einem höheren Niveau als die externen Deklarationen vorkommen, welche in dem eingegebenen Vorlagendatenstrom angetroffen worden sind, so daß ein Suchen in einer LIFO-(last in to first out) Anordnung vorkommt.  
 10. Verfahren zum Steuern der Darstellung einer Vorlage, welche durch einen Vorlagendatenstrom festgelegt ist, dadurch gekennzeichnet, daß  
 a) in eine Prozessoreinheit ein Datenstrom eingegeben wird, welcher eine darzustellende Vorlage festlegt, wobei der Datenstrom Seitensatz-Definitionen, wobei jede Seitensatzdefinition einen Anfang, null oder mehr Bildelemente und ein Ende umfaßt, wobei die Bildelemente null oder mehr Prologe aufweisen, wobei jeder Prolog null oder mehr Prologelemente aufweist, null oder mehr externe Datendeklarationen und Inhaltsteile einschließt, wobei die Inhaltsteile bilddefinierende Tokenfolge-Elemente enthaltender eingegebene Vorlagendatenstrom einer automatischen Syntax-Analyse (arsing) unterzogen wird, um den Anfang und das Ende eines Prologs in dem eingegebenen Vorlagendatenstrom festzulegen, um dessen Existenz zu bestimmen und um die Existenz irgendwelcher externer Datendeklarationen zu bestimmen;  
 b) ein Speicher mit einem vorgegebenen hierarchischen Pegel einer Speichertzuteilung zugeordnet wird, in dem zugeordneten Speicher



- bei den Vorlagendaten mit dem vorgegebenen, hierarchischen Pegel einen Hinweis auf die Stelle einer externen Datendeklaration ordnen und speichern, welche einem vorgegebenen Prologabschnitt zugeordnet ist, wodurch die externe Datendeklaration mit einem vorgegebenen Prologpegel zugeordnet wird, so daß bei irgendeinem weiteren Hinweis auf den vorgegebenen Prolog der Hinweis zu der externen Deklaration eingebunden wird;
- c) auf einen Stapel ein Zeiger hinweist, welcher einen Speicher identifiziert, der bei dem vorgegebenen hierarchischen Pegel zugeordnet worden ist;
- d) ein Speicher mit einem niedrigeren hierarchischen Pegel unter dem vorgegebenen hierarchischen Pegel zugeordnet wird und an den Vorlagendaten mit dem niedrigeren hierarchischen Pegel ein Hinweis auf die (Speicher-) Stelle einer externen Datendeklaration gespeichert wird, die einem vorgegebenen Prologabschnitt zugeordnet ist, wobei der externen Deklaration ein vorgegebener Prologpegel zugeordnet ist, so daß bei irgendeinem weiteren Hinweis auf den gegebenen Prolog der Hinweis auf die externe Deklaration eingebunden wird, und
- e) auf den Stapel ein Zeiger hinweist, welcher einen Speicher kennzeichnet, welcher dem niedrigeren hierarchischen Pegel zugeteilt ist.
11. Prozessor zum Steuern einer Darstellung einer Vorlage, welche durch einen Vorlagendatenstrom definiert ist, wobei der Prozessor programmiert ist, um die folgenden Schritte durchzuführen, daß
- a) in einen Prozessor ein Vorlagendatenstrom eingegeben wird, welcher eine darzustellende Vorlage festlegt, wobei der Vorlagendatenstrom Seitensatz-Definitionen, wobei jede Seitensatzdefinition einen Anfang, null oder mehr Bildelemente und ein Ende einschließt, wobei die Bildelemente null oder mehr Prologe einschließen und jeder Prolog null oder mehr Prologelemente aufweist, null oder mehr externe Datendeklarationen und Inhaltsteile einschließt, wobei die Inhaltsteile bilddefinierende Tokenfolge-Elemente enthalten;
- b) der eingegebene Vorlagen-Datenstrom einer automatischen Syntax-Analyse (Parsing) unterzogen wird, um den Anfang und das Ende eines Prologs in dem eingegebenen Vorlagendatenstrom festzulegen, um dessen Existenz zu bestimmen und um die Existenz irgendwelcher externer Datendeklarationen festzulegen;
- c) ein Speicher bei einem vorgegebenen hierarchischen Speicherzuteilungspegel zugeteilt wird und in dem zuge teilten Speicher mit Vorlagendaten mit vorgegebenem hierarchischem Pegel ein Hinweis auf die (Speicher-) Stelle einer externen Datendeklaration geordnet und gespeichert wird, welche einem vorgegebenen Prologabschnitt zugeordnet ist, wobei die externe Datendeklaration mit einem vorgegebenen Prologpegel zugeordnet ist, so daß bei irgendeinem weiteren Hinweis auf den vorgegebenen Pegel der Hinweis auf die externe Deklaration eingebunden wird;
- d) auf einen Stapel ein Zeiger hinweist, welcher einen Speicher kennzeichnet, welcher bei

- dem vorgegebenen hierarchischen Pegel zugeordnet ist;
- e) ein Speicher bei einem niedrigeren, hierarchischen Pegel unter dem vorgegebenen, hierarchischen Pegel zugeteilt wird und bei den Vorlagendaten mit dem niedrigeren hierarchischen Pegel ein Hinweis auf die (Speicher-) Stelle einer externen Daten-deklaration gespeichert wird, die einem vorgegebenen Prologabschnitt zugeordnet ist, wobei die externe Deklaration mit einem vorgegebenen Prologpegel zugeordnet wird, so daß bei irgendeinem weiteren Hinweis auf den gegebenen Prolog der Hinweis auf die externe Deklaration eingebunden wird, und
- f) auf den Stapel ein Zeiger hinweist, welcher den Speicher identifiziert, welcher bei dem niedrigeren hierarchischen Pegel zugeordnet ist.
12. Computer-lesbares Medium zum Speichern eines Programms zum Steuern einer Darstellung einer Vorlage, welche durch einen Vorlagendatenstrom festgelegt ist, wobei das Programm die Schritte festlegt, daß
- a) in eine Proessoreinheit ein Vorlagendatenstrom eingegeben wird, welcher eine darzustellende Vorlage festlegt, wobei der Vorlagendatenstrom Seitensatz-Definitionen, wobei jede Seitensatz-Definition einen Anfang, null oder mehr Bildelemente und ein Ende einschließt, wobei die Bildelemente null oder mehr Prologe aufweisen, und jeder Prolog null oder mehr Prologende, aufweist, null oder mehr externe Datendeklarationen und Inhaltsteile einschließt, wobei die Inhaltsteile bilddefinierende Tokenfolge-Elemente enthalten;
- b) der eingegebenen Vorlagendatenstrom einer automatischen Syntax-Analyse (Parsing) unterzogen wird, um den Anfang und das Ende eines Prologs in dem eingegebenen Vorlagendatenstrom zu bestimmen, um dessen Existenz festzulegen und um die Existenz irgendwelcher externer Datendeklarationen festzulegen;
- c) eine Speicher bei einem vorgegebenen hierarchischen Speicherzuordnungspegel zugeordnet wird, in dem zugeordneten Speicher an den Vorlagendaten mit einem vorgegebenen hierarchischen Pegel ein Hinweis auf die (Speicher-)Stelle einer externen Datendeklaration angeordnet und gespeichert wird, welche einem gegebenen Prologabschnitt zugeordnet ist, welcher dadurch die externe Datendeklaration mit einem vorgegebenen Dialogpegel zugeordnet, so daß bei irgendeinem weiteren Hinweis auf den vorgegebenen Prolog der Hinweis auf die externe Deklaration eingebunden wird;
- d) auf einen Stapel ein Zeiger hinweist, welcher einen Speicher kennzeichnet, der bei dem vorgegebenen hierarchischen Pegel zugeordnet worden ist;
- e) ein Speicher mit einem niedrigeren hierarchischen Pegel unter dem vorgegebenen hierarchischen Pegel zugeordnet wird und bei den Vorlagendaten mit dem niedrigeren hierarchischen Pegel ein Hinweis auf die (Speicher-)Stelle einer externen Datendeklaration ge-

speichert wird, welche einem vorgegebenen Prologabschnitt zugeordnet ist, der dadurch die externe Deklaration mit einem gegebenen Prologpegel zuordnet, so daß bei irgendeinem weiteren Hinweis auf den vorgegebenen Prolog der Hinweis auf die externe Deklaration mit eingebunden ist, und  
f) auf den Stapel ein Zeiger hinweist, welcher einen Speicher kennzeichnet, der bei dem niedrigeren hierarchischen Pegel zugeordnet ist.

---

Hierzu 8 Seite(n) Zeichnungen

---

15

20

25

30

35

40

45

50

55

60

65

- Leerseite -

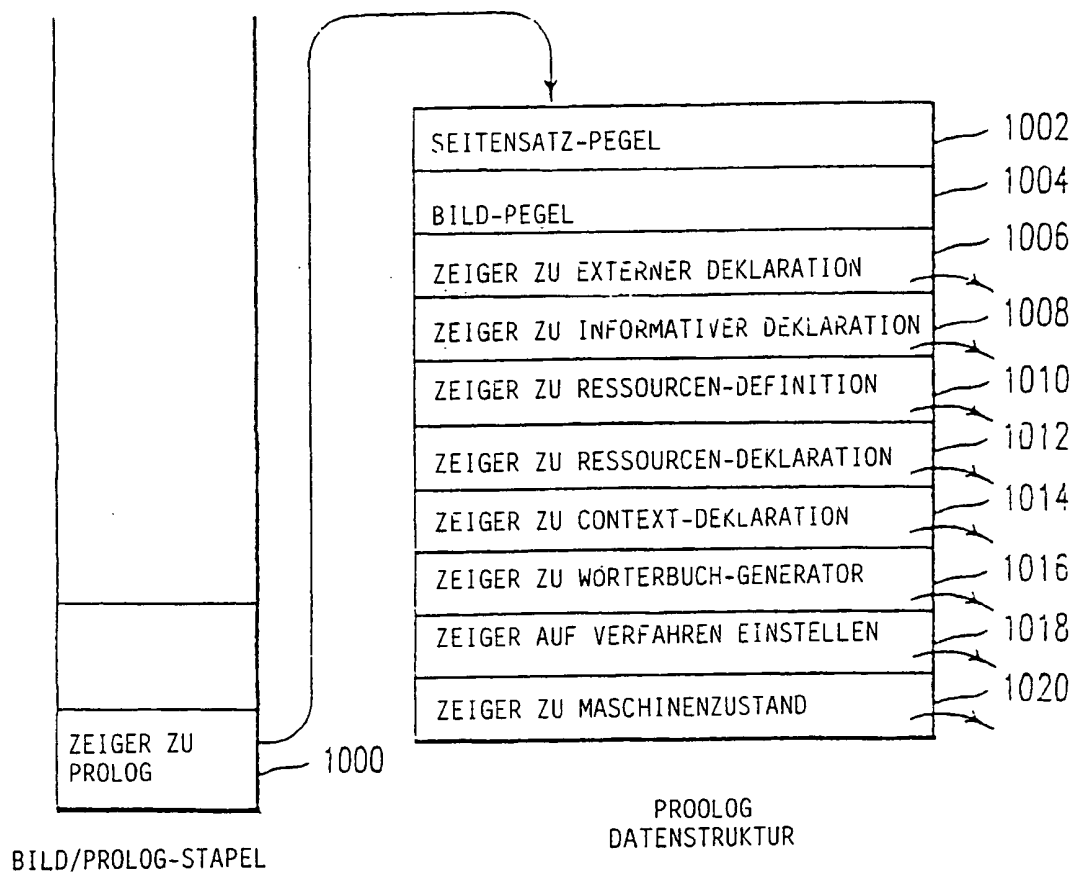
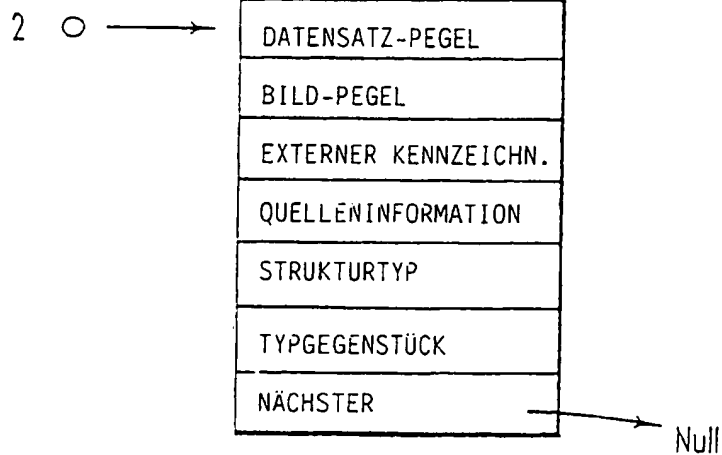


FIG. 1

1 ○ —————> Null

EXTERNE DEKLARATIONS-DATENSTRUKTUR



EXTERNE DEKLARATIONS-DATENSTRUKTUR

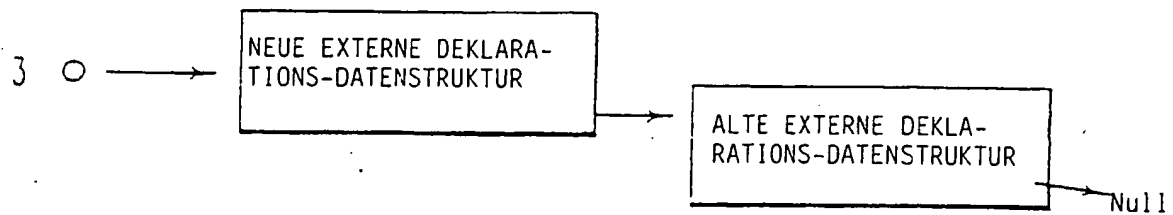


FIG. 2

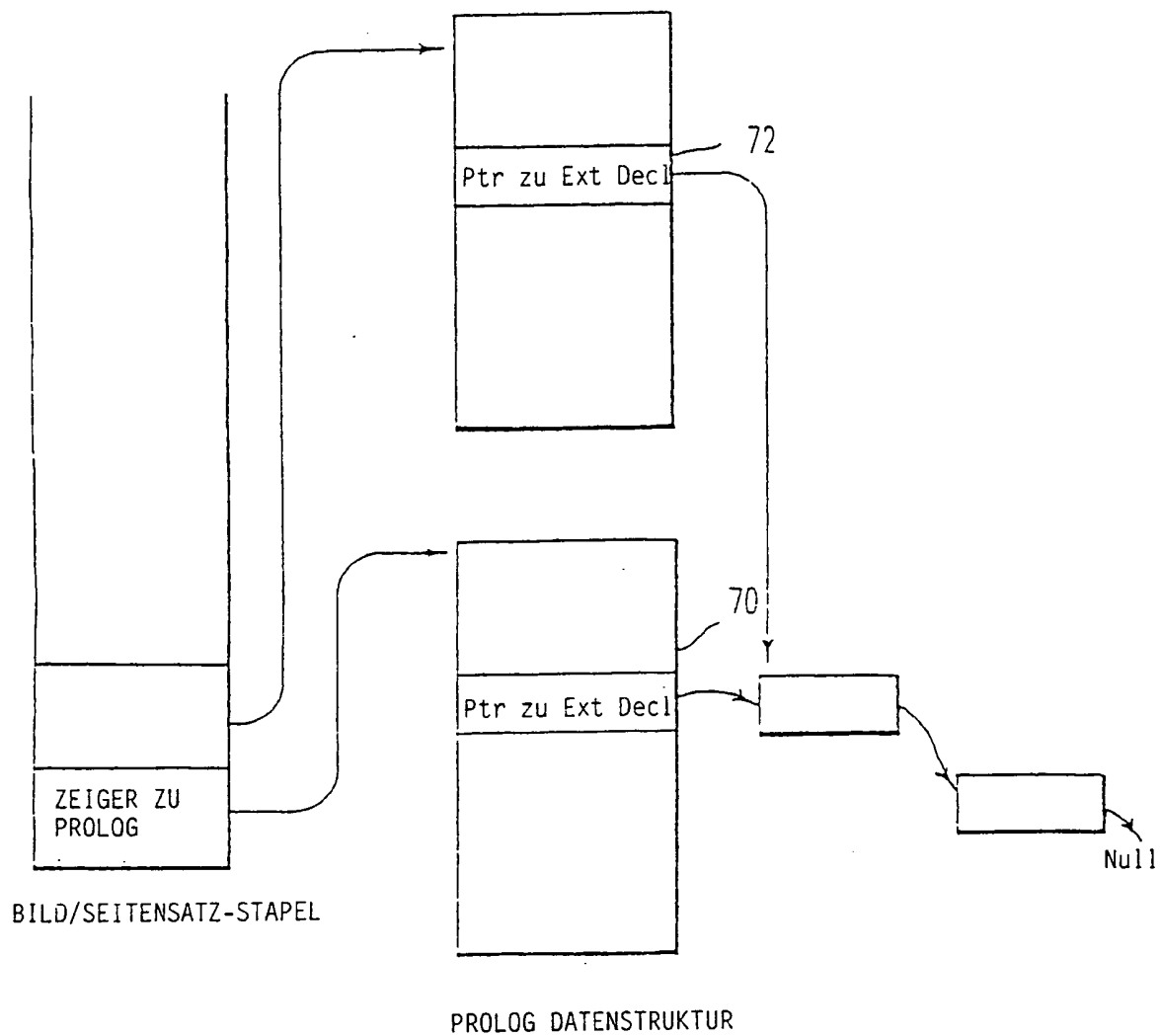
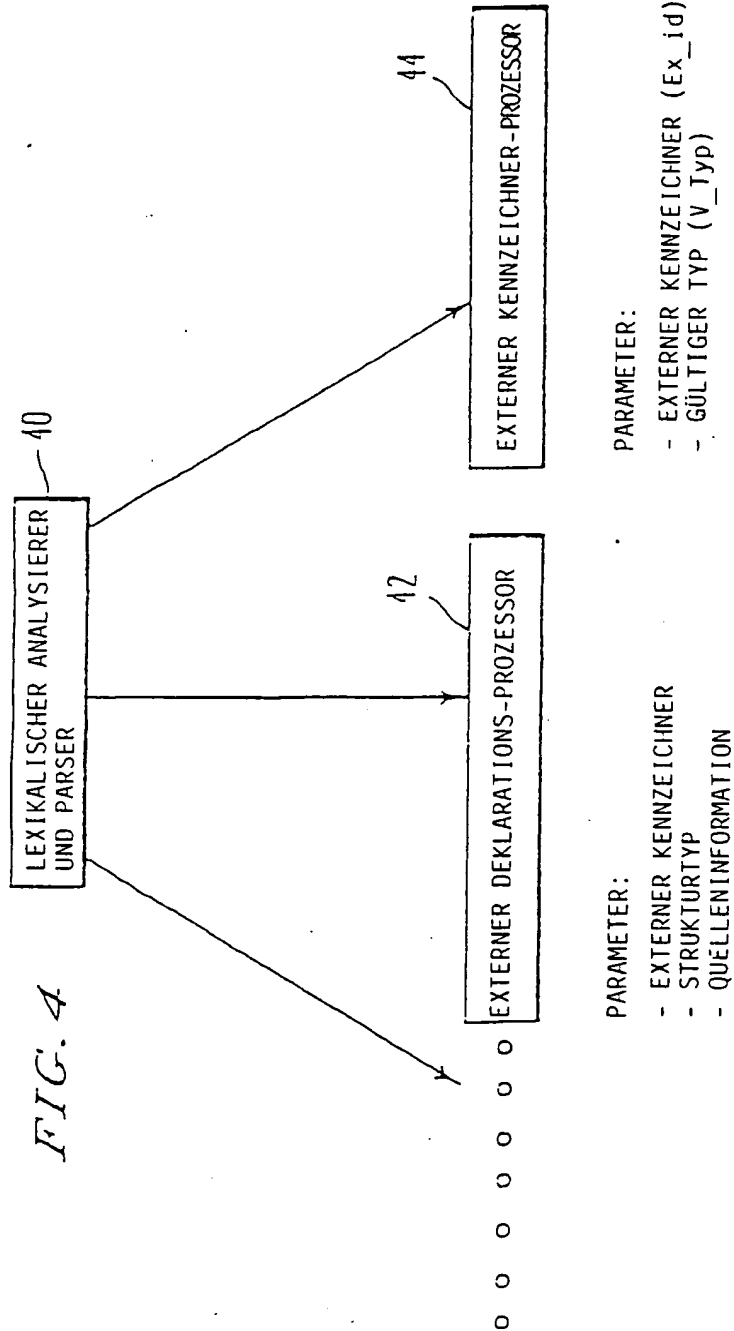


FIG. 3

FIG. 4



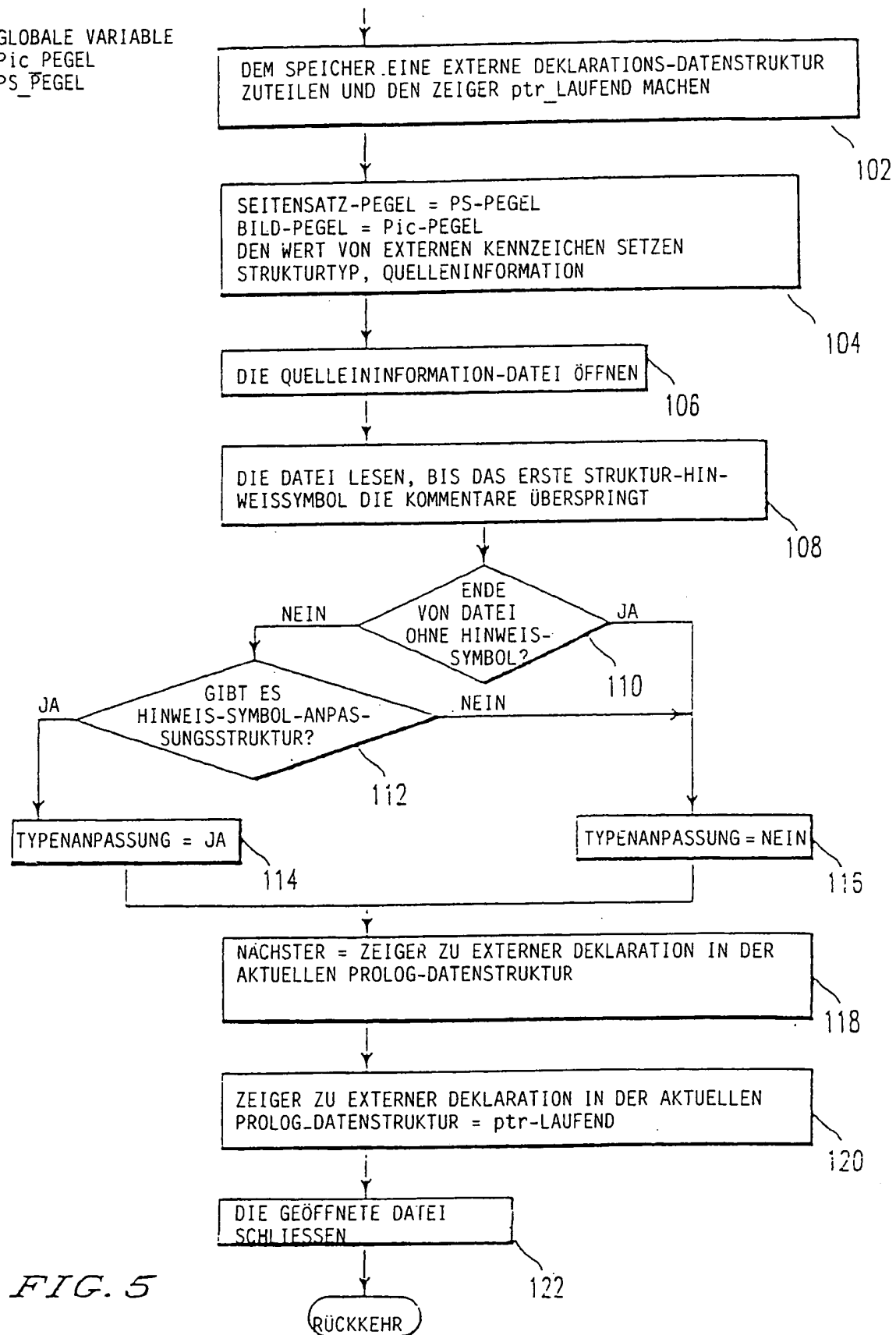
GLOBALE VARIABLE  
Pic\_PEGEL  
PS\_PEGEL

FIG. 5



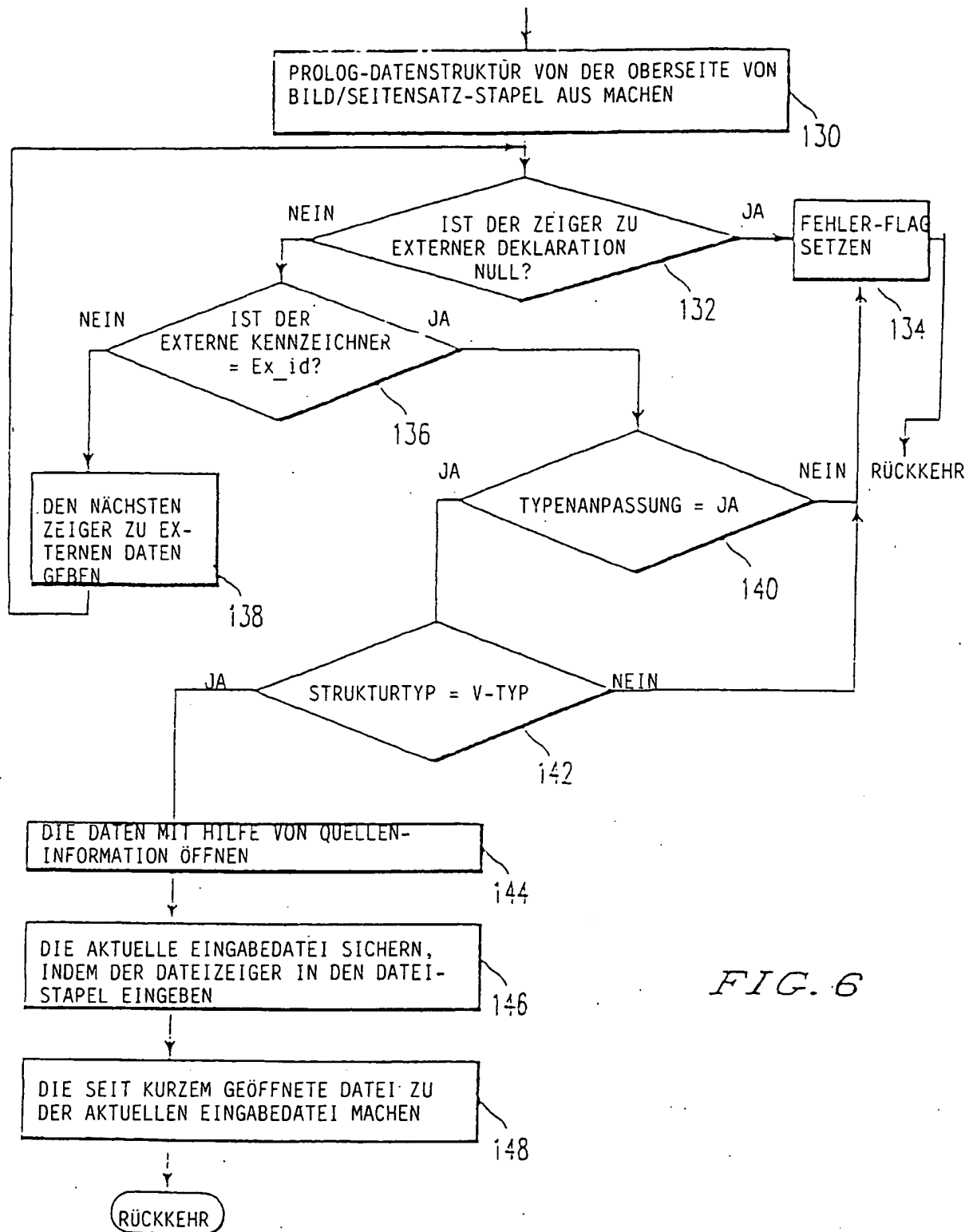


FIG. 6

GLOBALE VARIABLE: AKTUELLE EINGABEDATEI IN  
DATEISTAPEL EINGEBEN

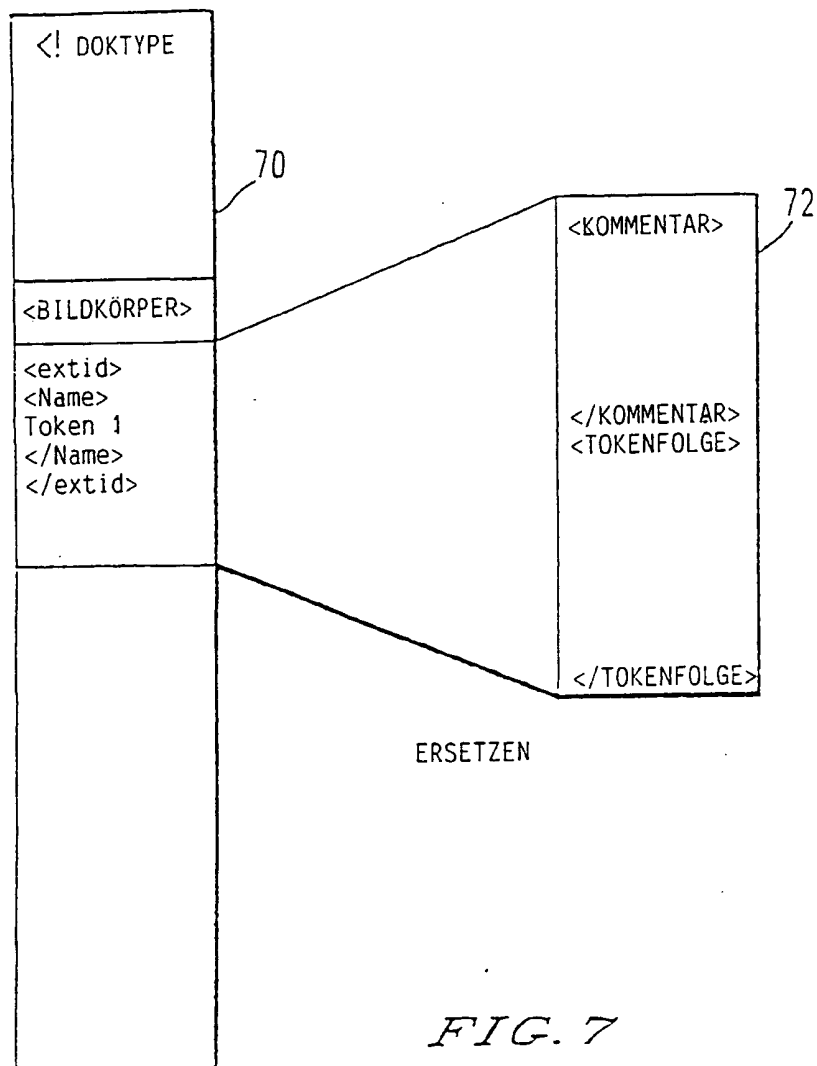


FIG. 7

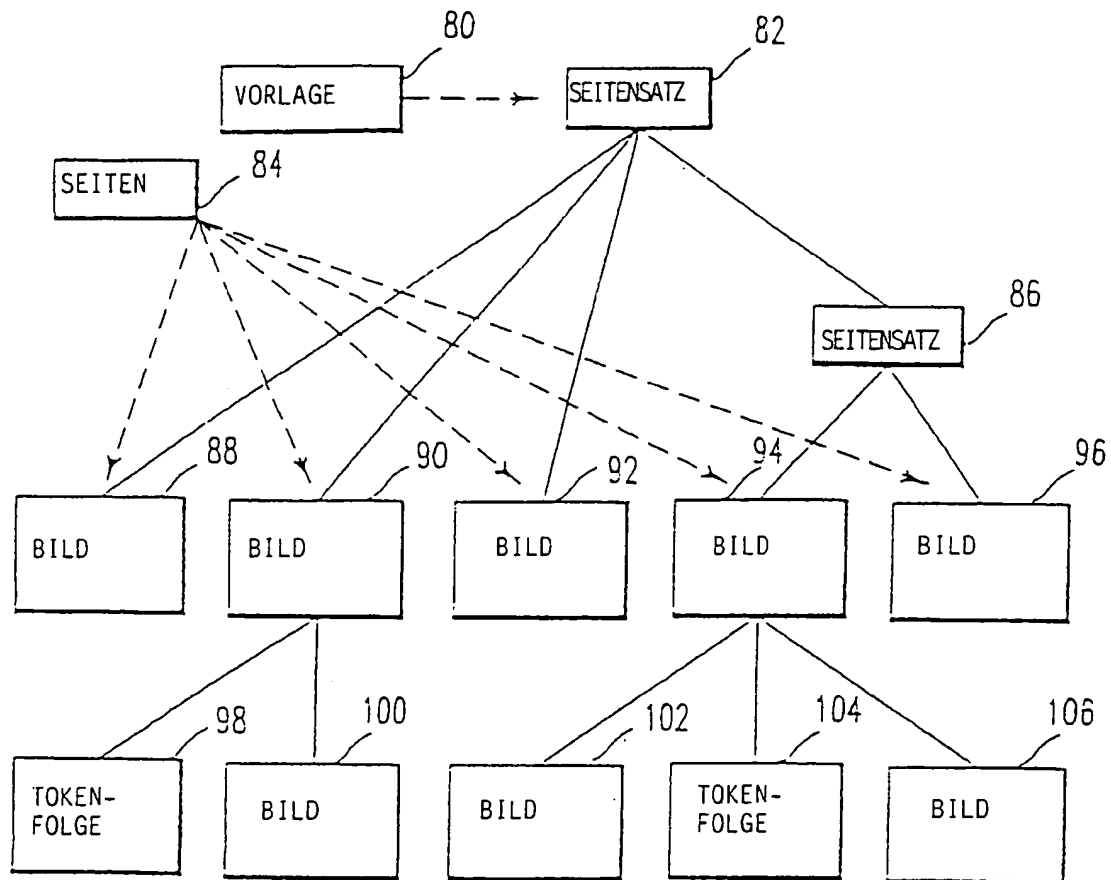


FIG. 8

**THIS PAGE BLANK (USPTO)**